Processing Multi-Touch Events on iPhone

Session 103

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Processing Multi-Touch Events on iPhone

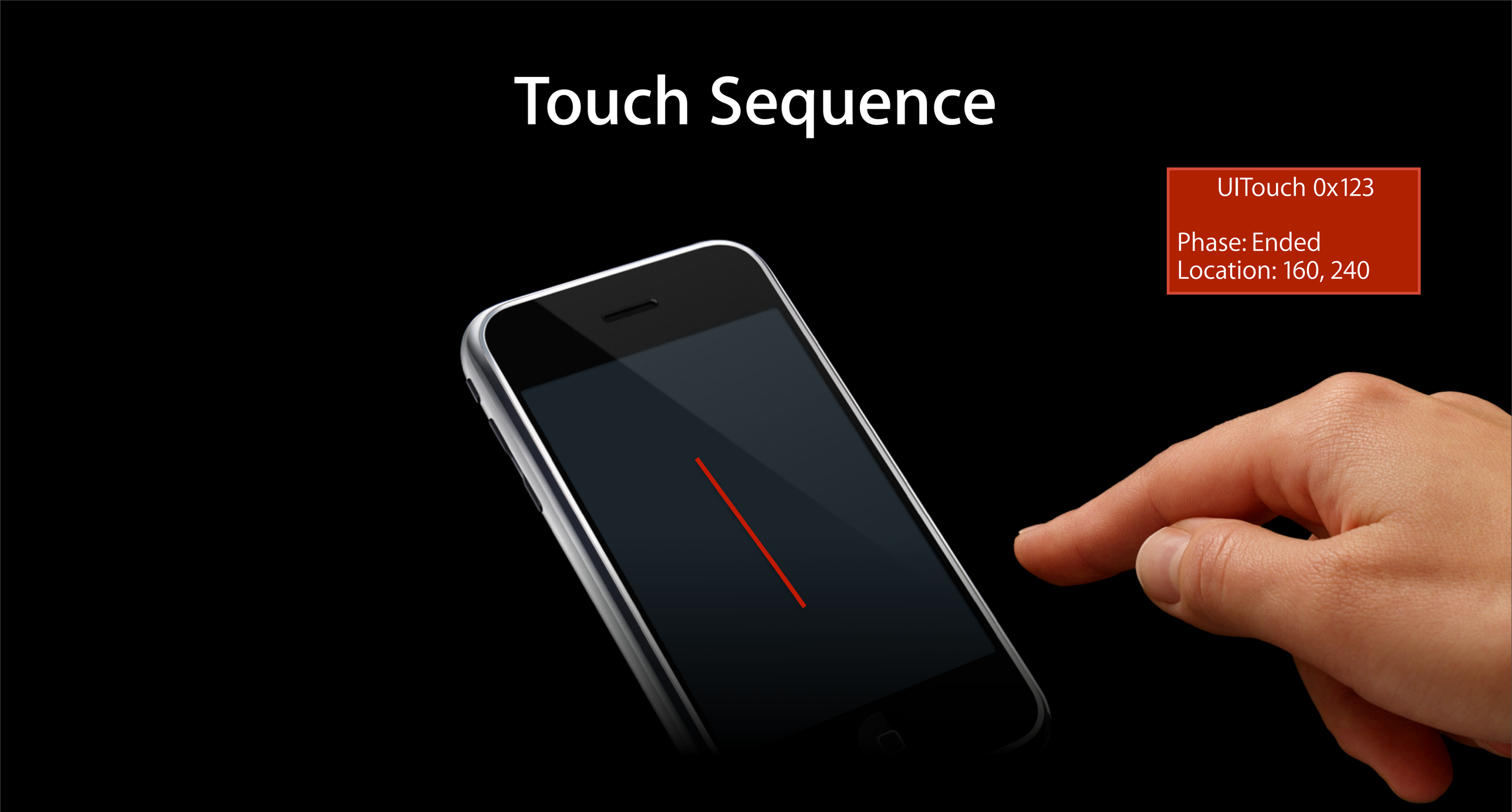
Josh Shaffer and Jason Beaver
iPhone Frameworks Engineers

Overview

- Building blocks—UITouch, UIEvent, UIResponder, and UIControl
- Event types—new in iPhone OS 3.0
- Encapsulation—touches as gestures
- Best practices
- Gotchas

Single-Touch Sequence

Touch Sequence



touchesBegan:(NSSet*)touches withEvent:(UIEvent*)event

touchesMoved:(NSSet*)touches withEvent:(UIEvent*)event

Touch Cancellation

UITouch 0x123

Phase: Cancelled
Location: 160, 240

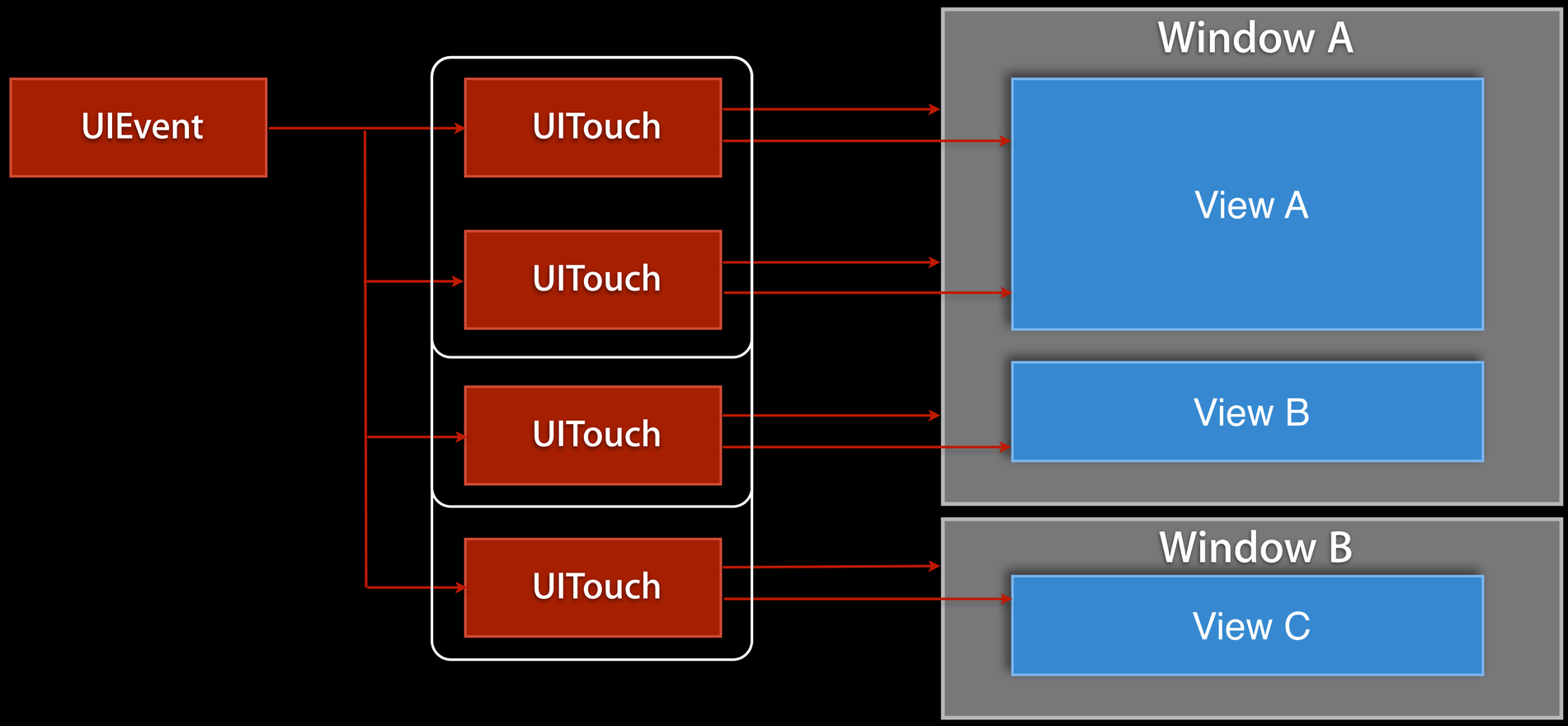
touchesBegan:withEvent:

touchesMoved:withEvent:

touchesEnded:withEvent:

touchesCancelled:withEvent:

UIEvent

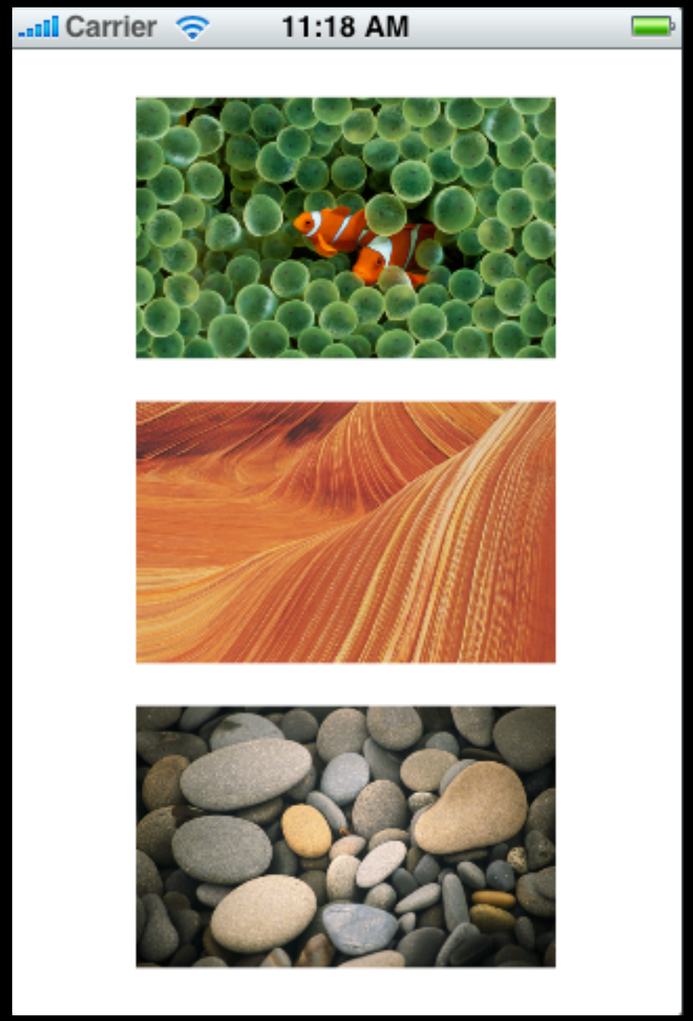



```
- (NSSet *)allTouchesForWindow:(UIWindow *)window;
- (NSSet *)touchesForView:(UIView *)view;
```

UIControl

```
UIControlEventTouchDown
UIControlEventTouchDownRepeat
UIControlEventTouchDragInside
UIControlEventTouchDragOutside
UIControlEventTouchDragEnter
UIControlEventTouchDragExit
UIControlEventTouchUpInside
UIControlEventTouchUpOutside
UIControlEventTouchCancel
```

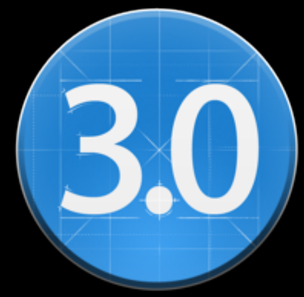
Multi-Touch Demo



Demo

Events

Event Types




```
UIEventType type;

  UIEventTypeTouches
  UIEventTypeMotion

UIEventSubtype subtype;

  UIEventSubtypeNone
  UIEventSubtypeMotionShake
```

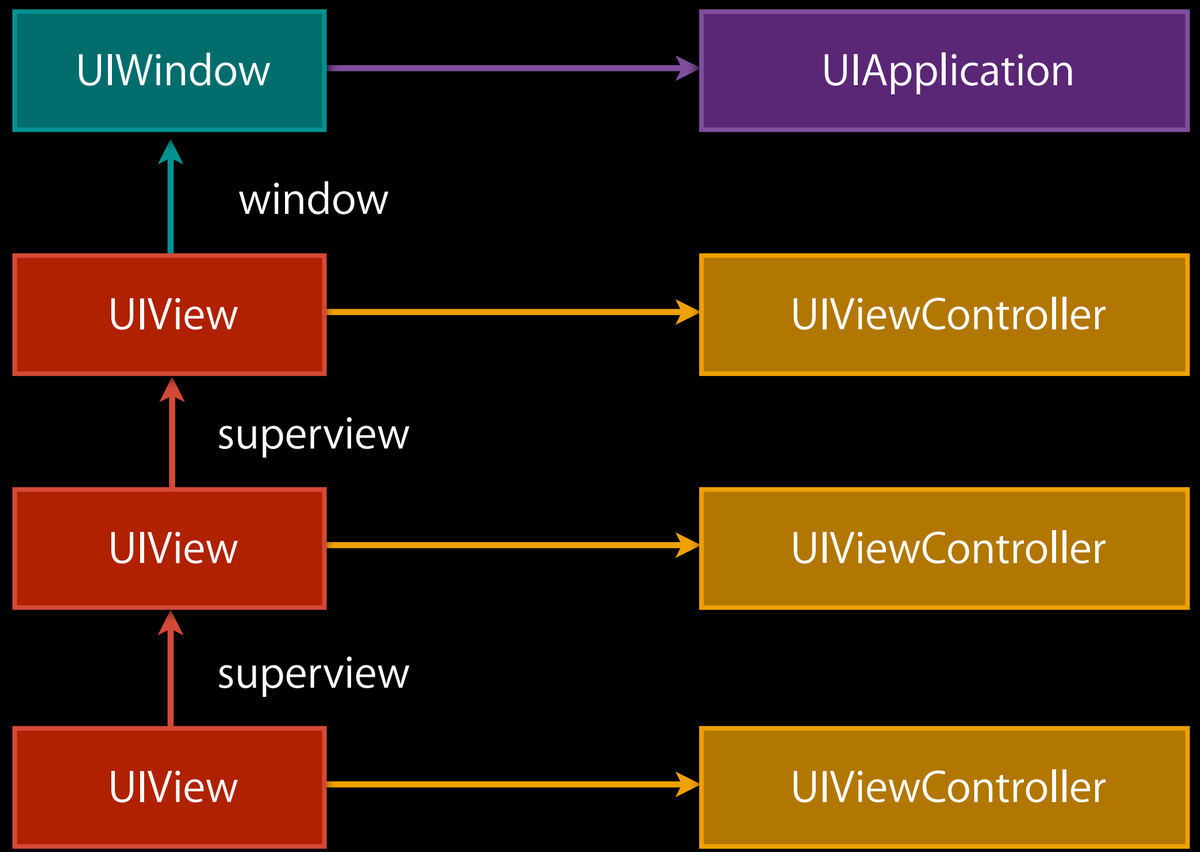
Receiving Events

UIResponder and you

```
UIEventTypeTouches
    -(UIView*)hitTest:(CGPoint)point
        withEvent:(UIEvent*)event

UIEventTypeMotion
```

Responder Chain



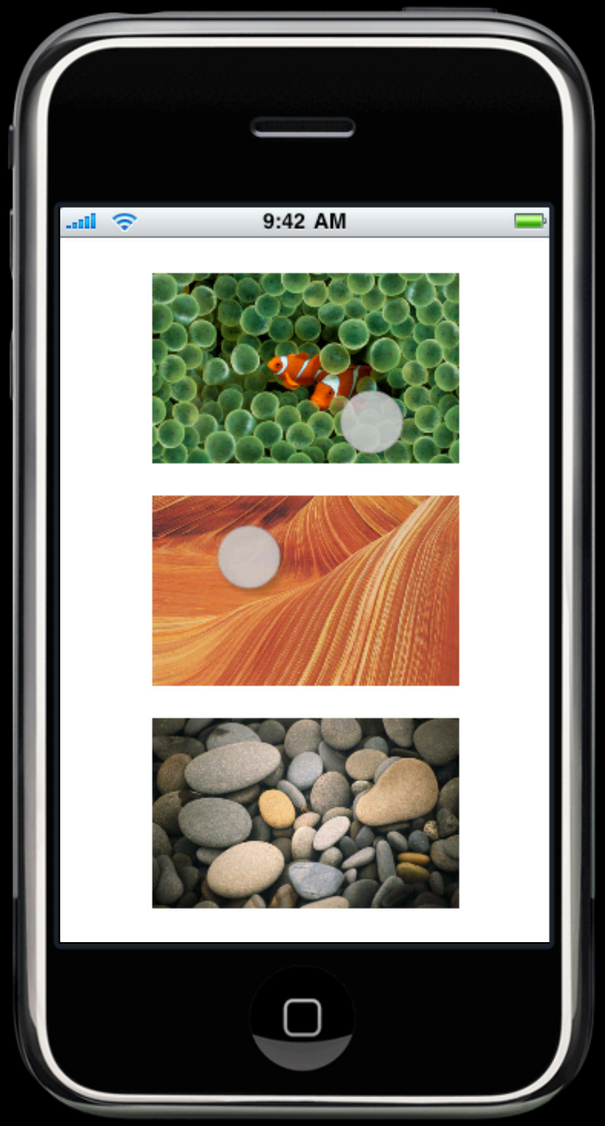
Gestures

Single Touch

Multiple Touch

Multiple Touch

Multiple Touch



UIResponder

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
```

UIResponder

```
UIWindow: - (void)sendEvent:(UIEvent*)event;
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
```

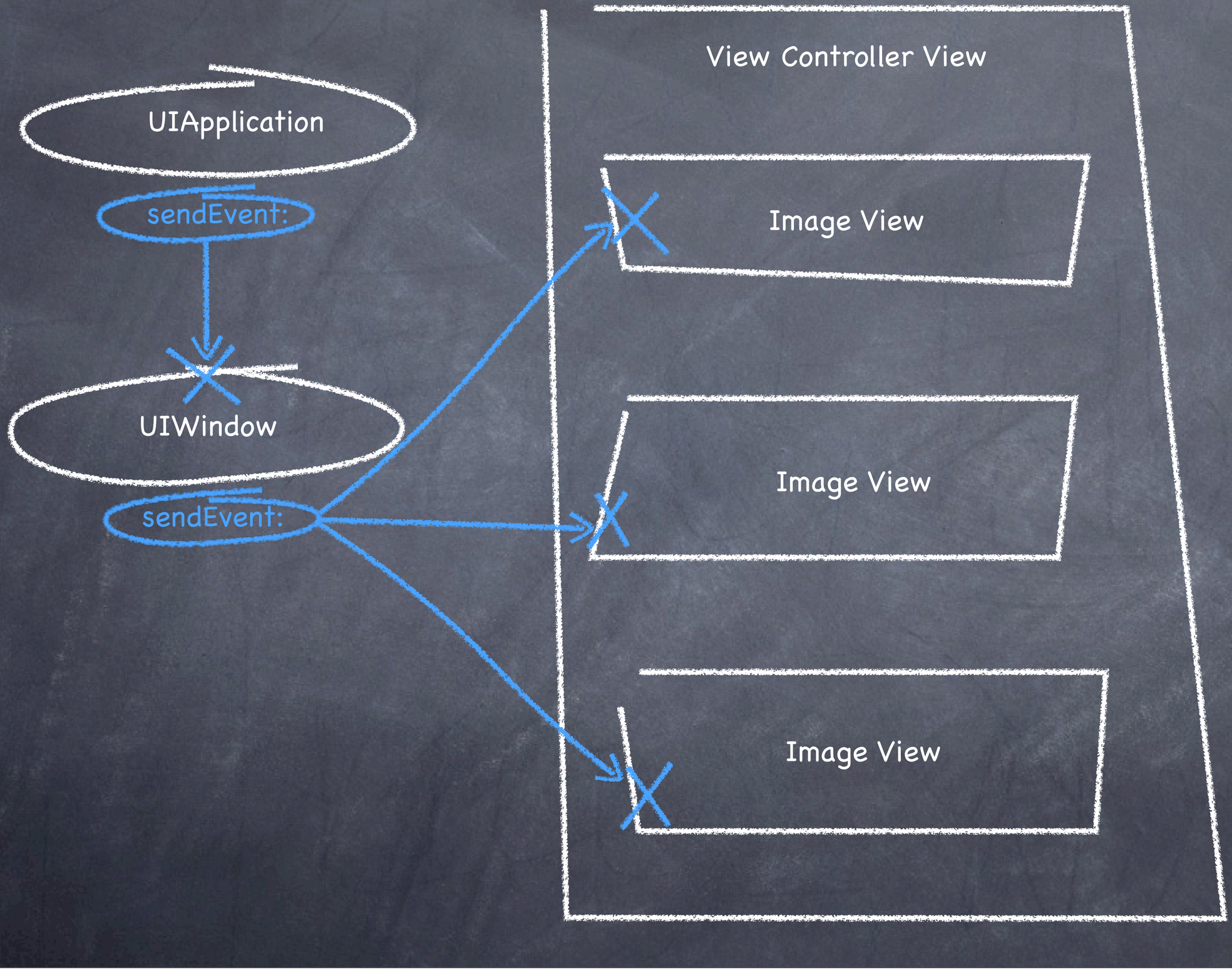
UIResponder

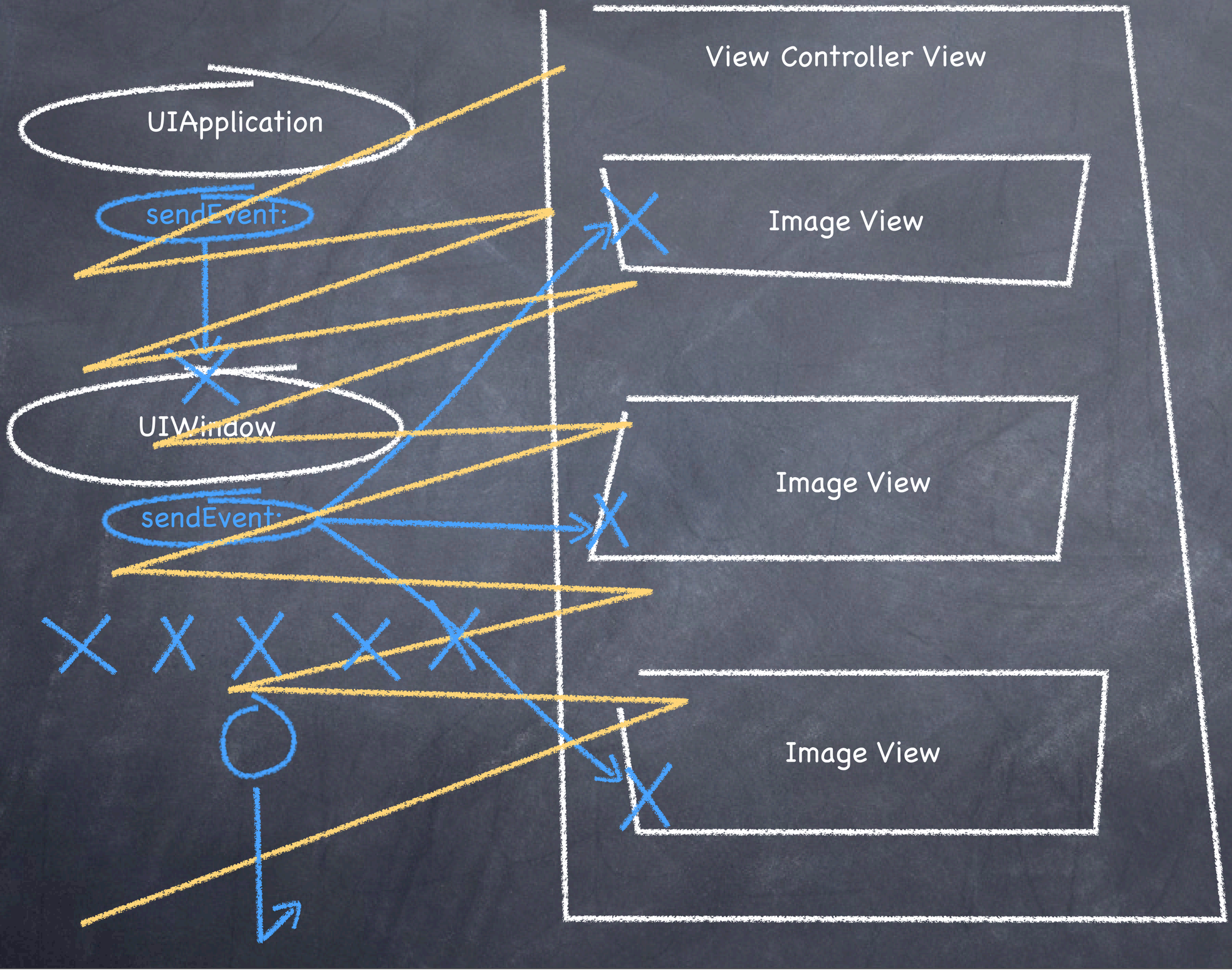
```
UIApplication: - (void)sendEvent:(UIEvent*)event;

   UIWindow: - (void)sendEvent:(UIEvent*)event;

- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
```

Always call `[super sendEvent:]`!


UIApplication
sendEvent:
UIWindow
sendEvent:
View Controller View
Image View
Image View
Image View


View Controller View
UIApplication
sendEvent:
Image View
UIWindow
sendEvent:
Image View
Image View

UIApplication

UIWindow

View Controller View

Image View

Transform Gesture

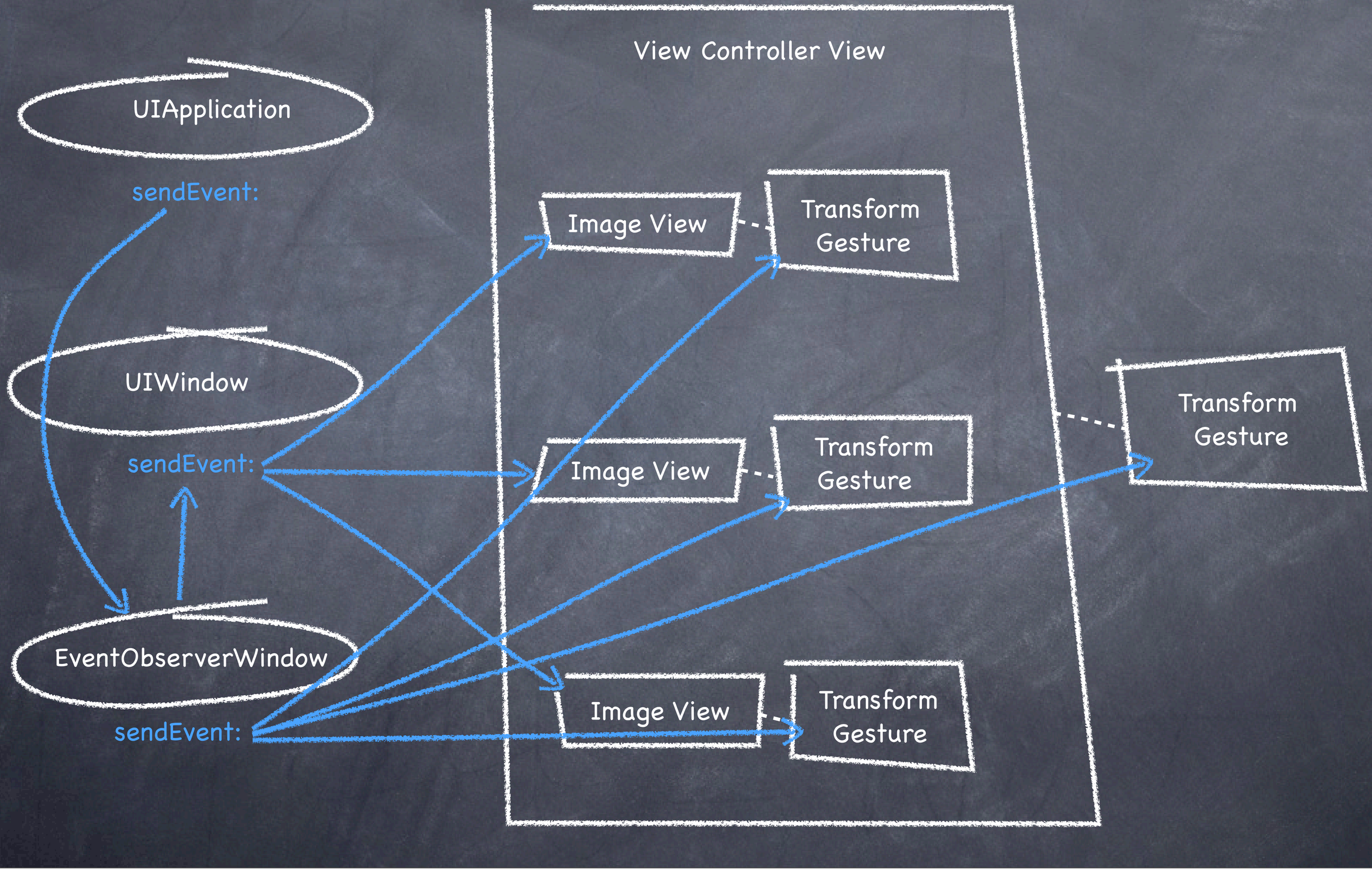
Image View

Transform Gesture

Image View

Transform Gesture

Transform Gesture


View Controller View
UIApplication
sendEvent:
UIWindow
sendEvent:
EventObserverWindow
sendEvent:
Image View
Transform Gesture
Image View
Transform Gesture
Image View
Transform Gesture
Transform Gesture

Demo

Best Practices

Handling Touch Events

The view returned by:

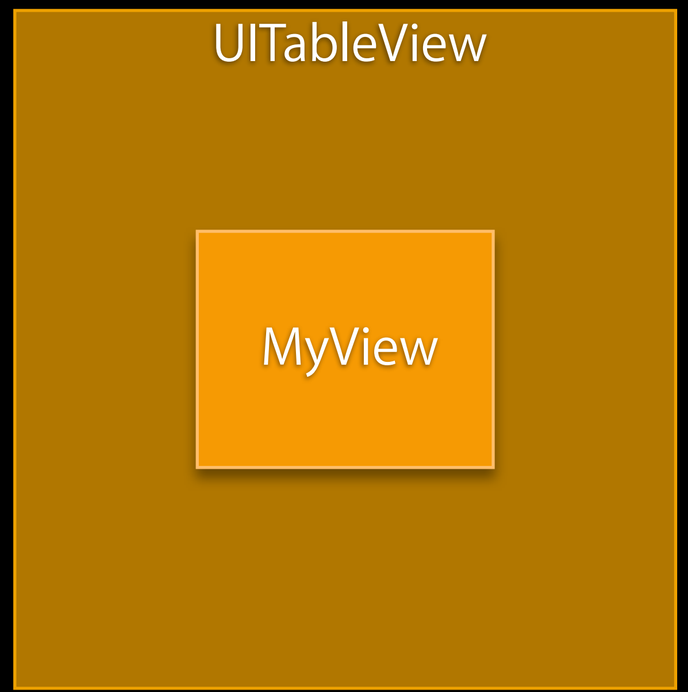
```
- (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event;
```

must handle **all** of the touch processing methods

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
```

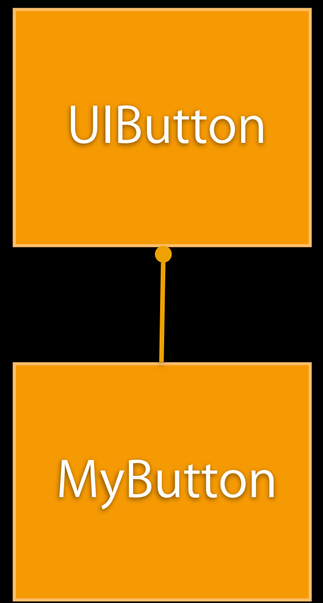
Subclassing UIView

Subclasses of UIView must implement all touch processing methods and must not call super



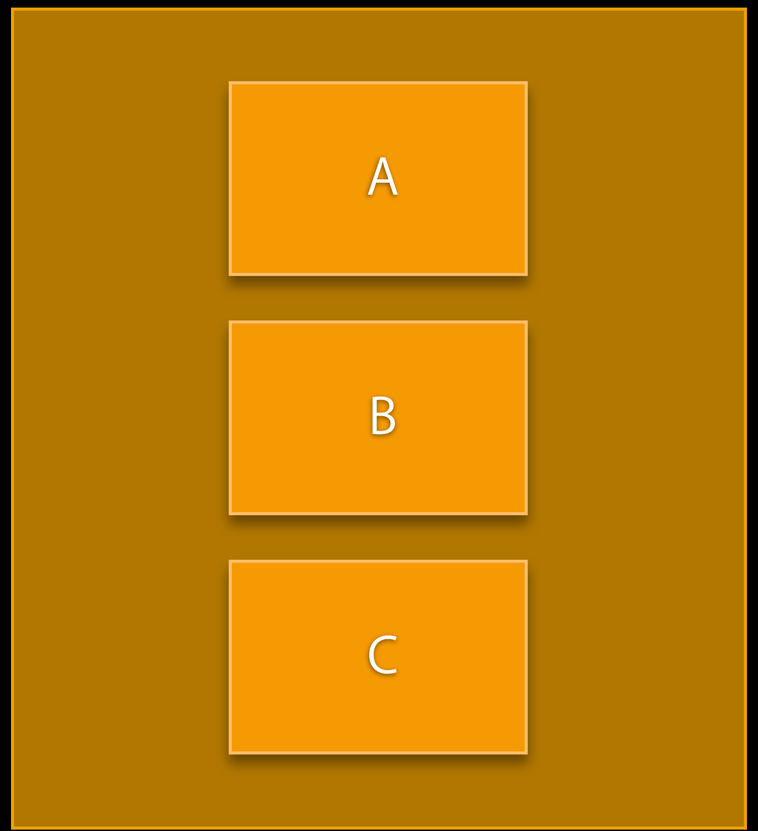
Subclassing Other UIKit Classes

Subclasses of any other UIKit class can implement any or all touch processing methods but must call super



Touch Forwarding

If you need to conditionally send touches to various views, all of the views involved need to be your own custom subclasses of UIView



Gotchas

Gotchas

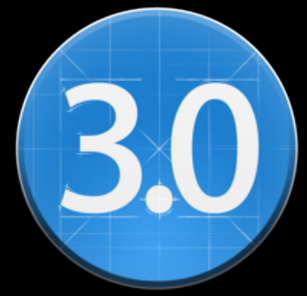
- Failing to handle cancelled touches

```
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
```

- Forwarding touches to UIKit classes
- Subclassing a UIKit class and conditionally calling super from the various touch processing methods
- Calling the wrong touch handler on your superclass

Motion Events

Receiving Motion Events




```
- (void)motionBegan:(UIEventSubtype)motion withEvent:(UIEvent *)event;
- (void)motionEnded:(UIEventSubtype)motion withEvent:(UIEvent *)event;
- (void)motionCancelled:(UIEventSubtype)motion withEvent:(UIEvent *)event;
```

Becoming First Responder

```
[responder becomeFirstResponder];

- (BOOL)canBecomeFirstResponder
{
    return YES;
}
```

Demo

Demo

More Information

Matt Drance
Application Frameworks Evangelist
mdrance@apple.com

Documentation
iPhone Application Programming Guide
http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide

Labs

iPhone Events and Gestures Lab	iPhone Lab A Thursday 2:00PM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09